Defense Threat Reduction Agency
8725 John J. Kingman Road, MS
6201 Fort Belvoir, VA 22060-6201

DTRA-TR-10-58

TECHNICAL REPORT

Robust Functionality and Active Data Management for Cooperative Networks in the Presence of WMD Stressors

Approved for public release; distribution is unlimited.

September 2011

HDTRA1-07-0036

Majeed Hayat et al.

Prepared by:
The Regents of the University of New Mexico
1 – University of New Mexico
Albuquerque, NM 87131

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-08-2009	DTRA Final	08/01/2007-06/30/2009

4. TITLE AND SUBTITLE
Robust Functionality and Active Data Management for Cooperative Networks in the Presence of WMD Stressors

5a. CONTRACT NUMBER

5b. GRANT NUMBER
HDTRA1-07-0036

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Majeed Hayat, Patrick G. Bridges, Yasamin Mostofi, and Patricia Crowley

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Regents of the University of New Mexico
1 – University of New Mexico
Albuquerque, NM 87131-0001

8. PERFORMING ORGANIZATION REPORT NUMBER
OVPRED 798B

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this project, we have begun the development of a rigorous probabilistic framework enabling new understanding and control of distributed networks' vulnerabilities to WMD-induced failures. We have developed a general stochastic queuing model and performed basic analysis to characterize the statistics of the task completion time treated as a random variable. Using the developed framework, we have established a theoretical/computational optimization tool that maximizes a network's robustness to node/link failures by constantly redistributing the network's computational loads. The framework also includes a rigorous treatment of the time evolution of networks subject to multiple random-in-space-and-time disruption and restoration events. Next, we have developed a theory to describe and solve the problem of a network attempting to reach consensus on the occurrence of a WMD attack based upon network data. To this end, we have developed the mathematical foundations of networked binary consensus over noisy links. Finally, we have considered real-time adaptation in distributed, unreliable networks in the face of the potentially correlated failures induced by WMD stressors, establishing methods for driving adaptation using low-overhead monitoring systems.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	None	24	Frank Gilfeather
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER *(Include area code)* 505-272-7039

FINAL REPORT
"Robust Functionality and Active Data Management for Cooperative Networks in the Presence of WMD Stressors"

Majeed M. Hayat[1], Patrick G. Bridges[1], Yasamin Mostofi[1], and Patricia Crowley[2]

In this project, we have begun the development of a rigorous probabilistic framework that has enabled new understanding and control of distributed networks' vulnerabilities to WMD-induced failures through a quantitative assessment of basic network-functionality metrics. We have developed a general stochastic queuing model and performed basic analysis to characterize the statistics of the task completion time treated as a random variable. Using the developed framework, we have established a theoretical/computational optimization tool that maximizes a network's robustness to node/link failures by constantly redistributing the network's computational loads. The framework also includes a rigorous treatment for describing the time evolution of the functionality of any network subject to multiple random-in-space-and-time disruption and restoration events. Next, we have developed a theory to describe and solve the problem of a network attempting to reach consensus on the occurrence of a WMD attack based upon network data. To this end, we have developed the mathematical foundations of networked binary consensus over noisy links. Finally, we have considered real-time adaptation of computation and storage in distributed, unreliable (including sensor) networks in the face of significant, potentially correlated failures induced by WMD stressors: we have established methods for low-overhead monitoring systems that can drive decentralized adaptation.

In what follows we described the details of the basic research accomplished in this project as summarized above. We divide the description into four categories: (i) Probabilistic Network Modeling and Resilient Task Reallocation, (ii) Probabilistic Network Modeling of Network Functionality, iii) Network Health Assessment; and (iv) Active Data Management. Participants in these activities, including students, are listed in the respective sections below.

1. Probabilistic Network Modeling and Resilient Task Reallocation

Executive Summary

A rigorous probabilistic framework to analytically characterize the execution time of workloads in distributed computing systems (DCSs), subject to stochastic topological changes due to WMD attacks was developed as a result of this effort. The developed characterization considered a group of heterogeneous and geographically dispersed computing nodes, uncertainties in the communication network due to random non-negligible delays and stochastic, long-term node failures due to WMD attacks. The metric employed to assess the performance of the DCS is the service reliability, which was defined as the probability of executing a workload before all the computing nodes fail. Resilient task reallocation policies were obtained by solving a constrained optimization problem whose cost function employs the rigorous model developed for the service reliability of workloads. An algorithm that scales linearly with the number of nodes was also derived to reduce the computing complexity of the optimization problem. The mathematical model was validated using Monte-Carlo (MC) simulations and experimental data collected from a testbed DCS.

[1]The University Of New Mexico, Albuquerque, NM
[2]Gonzaga University, Spokane, WA

Personnel

The following personnel worked on problems related to this portion of the grant:

- Graduate Students: Jorge Pezoa, Zhuoyao Wang and Manuel Rivera
- Research Faculty: David Dietz
- Faculty: Majeed Hayat (PI)

Publications

- J. E. Pezoa, S. Dhakal and M. M. Hayat, "Decentralized Load Balancing for Improving Reliability in Heterogeneous Distributed Systems." In *Proc. of The International Workshop on Design, Optimization and Management of Heterogeneous Networked Systems (DOM-HetNetS '09), Vienna, Austria, September 22-25, 2009.*
- J. E. Pezoa, S. Dhakal and M. M. Hayat, "Maximizing service reliability in distributed computing systems with random failures: Theory and implementation," *IEEE Trans. Parallel and Distributed Systems*, accepted subject to minor revisions, 2009.
- M. M. Hayat. J. E. Pezoa, D. Dietz, and S. Dhakal, "Dynamic load balancing for robust distributed computing in the presence of topological impairments," *Wiley Handbook of Science and Technology for Homeland Security*, 2009.

Technical Summary

Introduction: DCSs offer a flexible, reliable, and powerful cooperative computing platform. When DCSs operate in harsh or threat-prone environments, factors such as limited or intermittent communication resources or long-term physical damage of the computing nodes, can result in random topological changes in the DCS, which, in turn, can severely degrade the performance and reliability of DCSs. From this, it is mandatory to develop control strategies for increasing the robustness of networks when a threat is present. In this report it is described how intelligent task reallocation strategies, and their mathematical stochastic models, can be exploited to increase the DCS's robustness to random topological changes, and simultaneously, how to use the available computing resources of the system efficiently, in the presence of communication uncertainty and node dysfunction inflicted by WMD attacks developed locally at UNM.

Mathematical model: In order to describe the reliability of a DCS in the presence of WMD attacks, we constructed a recursive model for the execution time of a workload served by a DCS [2,7,8]. The model constructed predicted accurately both the execution time of a workload and the reliability in executing a workload. The main assumption made in the development of the model is that all the random times governing the dynamics of the system are exponentially distributed. This assumption was key in order to obtain a tractable and computationally simple mathematical model for the reliability [2,7,8].

It was shown that, under the assumption of exponentially distributed random times, the configuration of a DCS can be described using the following quantities: (i) the number of tasks queued at each node; (ii) the functional or dysfunctional state of each node in the system; and (iii) the amount of tasks in transit over the communication network [2,7,8]. In order to mathematically describe at any time the state of an n-node DCS, we introduced in our analysis three state vectors: the system-queue, the system-function and the network state vector. The system-queue state vector, $\mathbf{M}(t)$, describes the number of tasks queued at each node in the system. The system-function vector, $\mathbf{F}(t)$, is a binary vector specifying the working or failed state of the nodes, while the network

state vector, $\mathbf{C}(t)$, describes the number of tasks being transferred to the nodes. In addition to the state vector, we introduced in our characterization for the execution time two parameters that define a task reallocation policy. The first parameter introduced was the reallocation instant, a non-negative real number, t_b, that specifies the instant when the task reallocation should be executed. The second parameter introduced was the reallocation strength, an n-by-n matrix $\mathbf{K}$ that specifies the number of tasks to reallocate between all pairs of nodes. These two parameters answer the following fundamental questions in distributed computing: (i) *When nodes have to execute the task reallocation policy?* (ii) *How many tasks have to be reallocated among the nodes?* and (iii) *Which nodes are appropriate to receive extra work from the other nodes?* In this effort, these parameters were determined after solving a constrained optimization problem that aims to maximize the service reliability of the DCS.

Next, the execution time of a workload, denoted by $T_{\mathbf{K}}(t_b;\mathbf{M}_0,\mathbf{F}_0,\mathbf{C}_0)$, was defined as the random time taken by the DCS to serve its entire workload if the task reallocation policy executed is as specified by t_b and $\mathbf{K}$, and the initial system configuration at $t=0$ is as specified by $\mathbf{M}_0=\mathbf{M}(0)$, $\mathbf{F}_0=\mathbf{F}(0)$ and $\mathbf{C}_0=\mathbf{C}(0)$. With this, the service reliability can be defined as the probability that all the tasks are served before all nodes fail, that is $R_{\mathbf{K}}(t_b;\mathbf{M}_0,\mathbf{F}_0,\mathbf{C}_0)\triangleq \mathsf{P}\{T_{\mathbf{K}}(t_b;\mathbf{M}_0,\mathbf{F}_0,\mathbf{C}_0)<\infty\}$. By exploiting the principle of stochastic regeneration, we showed that the service reliability of an n-node DCS satisfies the difference-differential equation (1) and the difference equation (2):

$$\frac{d}{dt_b}R_{\mathbf{K}}(t_b;\mathbf{M}_0,\mathbf{F}_0,\mathbf{C}_0)=\sum_{i=1}^{n}\lambda_{d_i}R_{\mathbf{K}}(t_b;\mathbf{M}_0-\boldsymbol{\Delta}^{ii},\mathbf{F}_0,\mathbf{C}_0)+\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\lambda_{ij}^{Q}R_{\mathbf{K}}(t_b;\mathbf{M}_0+\boldsymbol{\Delta}^{ji},\mathbf{F}_0,\mathbf{C}_0)$$
$$+\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\lambda_{ij}^{F}R_{\mathbf{K}}(t_b;\mathbf{M}_0^{ji},\mathbf{F}_0^{ji},\mathbf{C}_0)+\sum_{i=1}^{n}\sum_{j=1}^{g_i}\tilde{\lambda}_{j,i}R_{\mathbf{K}}(t_b;\mathbf{M}_0+f_{ii}\,l_{ji}\boldsymbol{\Delta}^{ii},\mathbf{F}_0,\mathbf{C}_0^{Z_{ji}})$$
$$+\sum_{i=1}^{n}\lambda_{f_i}R_{\mathbf{K}}(t_b;\mathbf{M}_0^{ii},\mathbf{F}_0^{ii},\mathbf{C}_0^{Y_i}),-\lambda R_{\mathbf{K}}(t_b;\mathbf{Q}_0,\mathbf{F}_0,\mathbf{C}_0) \tag{1}$$

$$R_{\mathbf{K}}(0;\mathbf{Q}_0,\mathbf{F}_0,\mathbf{C}_0)=\sum_{i=1}^{n}\frac{\lambda_{d_i}}{\lambda}R_{\mathbf{K}}(0;\mathbf{Q}_0-\boldsymbol{\Delta}_{ii},\mathbf{F}_0,\tilde{\mathbf{C}}_0)+\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}\frac{\lambda_{ij}^{F}}{\lambda}R_{\mathbf{K}}(0;\mathbf{Q}_0,\mathbf{F}_0^{ji},\mathbf{C}_0)$$
$$+\sum_{i=1}^{n}\sum_{j=1}^{g_i}\frac{\tilde{\lambda}_{j,i}}{\lambda}R_{\mathbf{K}}(0;\mathbf{Q}_0+f_{ii}\,l_{ji}\boldsymbol{\Delta}_{ii},\mathbf{F}_0,\mathbf{C}_0^{Z_{ji}})+\sum_{i=1}^{n}\frac{\lambda_{f_i}}{\lambda}R_{\mathbf{K}}(0;\mathbf{Q}_0^{ii},\mathbf{F}_0^{ii},\mathbf{C}_0^{Y_i}). \tag{2}$$

The model for the service reliability given in Equations (1) and (2) was employed to search for the optimal task reallocation instant, t_b^*, and the optimal task reallocation strength, $\mathbf{K}^*$, that maximizes the service reliability. This was performed by solving the constrained optimization problem

$$(t_b^*,\mathbf{K}^*)=\underset{(t_b,\mathbf{K})}{\operatorname{argmax}}\;\; R_{\mathbf{K}}(t_b;\mathbf{Q}_0,\mathbf{F}_0,\mathbf{C}_0) \tag{3}$$

subject to $t_b\geq 0$ and $K_{ij}\in[0,1]$.

Algorithm: Solving the optimization problem (3) is computationally expensive for DCSs with large number of nodes, as the amount of computations grows exponentially in the number of nodes. For example, it was shown that the complexity in solving Equation (1) is bounded by $\mathcal{O}(2^{n^2})$. To avoid such complication, an algorithm for devising task reallocation policies was developed, see Appendix A. The key idea was to decompose an n-node system into several two-node DCSs and

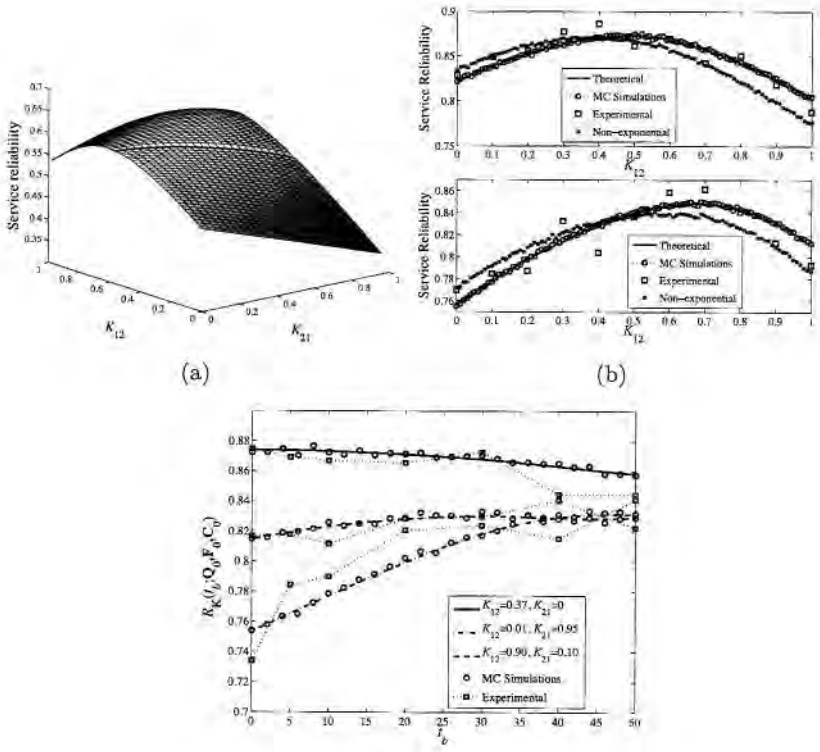


Figure 1. (a) Theoretical service reliability as a function of the task reallocation strengh $\mathbf{K}$ for a fixed reallocation instant. (b) Theoretical, simulated and experimental service reliability as a function of the task reallocation strengh of the node 1, when task reallocation is executed at $t_b = 0$. In the upper plot $K_{21} = 0.25$ while in the lower plot $K_{21} = 0.9$. (c) Service reliability as a function of the reallocation instant for three representative reallocation strenghts.

exploit the exact characterization for reliability of two-node systems provided by Equations (1) and (2). The algorithm developed was proven to scale linearly with the number of nodes in the DCS [8].

Key results: Evaluations for a two-node DCS were conducted in order to assess the performance of the reallocation policies devised by solving (3). Figure 1(a) shows the theoretical service reliability as a function of the reallocation strengths K_{12} and K_{21}. Figure 1(b) shows the theoretical, MC simulated and experimental service reliability as a function K_{12}, for a fixed value of K_{21}, where the last two parameters are the components of $\mathbf{K}$ and govern the intensity of load transfer between the two nodes. Notably, the theoretical and experimental results showed an excellent agreement. From the theoretical curves, it was observed that the service reliability seems to be a concave function of the number of tasks to be reallocated among the nodes, that is, a concave (reallocation strength parameter). Figure 1(c) shows the service reliability as a function of the reallocation instant, for three representative selections of the reallocation strength. It was observed again a remarkable agreement among theoretical prediction, Monte-Carlo (MC) simulation, and experimental results. From the figure, it was clearly observed that a proper choice of $\mathbf{K}$ resulted in an increase of the service reliability. It was noted also that an incorrect selection for the reallocation strength can be compensated by delaying the reallocation action.

The accuracy of the model in predicting the reliability was also assessed. Predictions generated by the model for reliability were compared with MC simulations where the distributions of the random times are non-exponential. Via MC simulations the service reliability of a non-Markovian

DCS was estimated with a 95% confidence. The estimated reliability is plotted in Figure 1(a). It was noted from the figure that the exponential model for reliability is very accurate and yields a relative approximation error below 4%. Further simulations showed that, as the ratio between the average transfer time and the average service time of the nodes increases, the exponential approximation looses its accuracy in predicting the reliability. Specifically, approximation errors of 120% were found when the ratio between the times is five.

Table 1. Service reliability achieved by three reallocation policies, which have different reallocation criteria. For comparison purposes, the optimal values obtained for at case are listed.

Initial load	Service reliability			
$(m_1, \ldots, m_5)$	Max-Service	Proc-Speed	Complete	Optimum
(150,0,0,0,0)	0.509	0.511	0.573	0.631
(0,150,0,0,0)	0.614	0.610	0.617	0.617
(0,0,150,0,0)	0.601	0.591	0.601	0.601
(0,0,0,150,0)	0.583	0.533	0.612	0.615
(0,0,0,0,150)	0.543	0.566	0.613	0.619
(30,30,30,30,30)	0.634	0.603	0.636	0.657
(59,2,4,34,51)	0.556	0.608	0.638	0.668
(18,55,29,27,21)	0.642	0.623	0.640	0.649
(26,30,28,38,28)	0.642	0.639	0.642	0.642
(40,15,40,35,20)	0.624	0.610	0.643	0.656

The effect of the selection of a reallocation criteria on the service reliability was studied in DCSs with multiple nodes. Three reallocation policies, each one of them having a different reallocation criterion, were studied. The Maximal-Service strategy reallocated taken when workload of the DCS is imbalanced with respect to the relative reliability of the nodes. The Processing-Speed strategy triggered a reallocation action when workload is imbalanced with respect to the relative processing rate of the nodes. Finally, the Complete reallocation strategy triggered a reallocation action when workload is imbalanced with respect to the combined processing and failure rates. The results of the evaluations conducted are listed in Table 1. The results listed in Table 1 showed that, in most of the cases, the three policies achieve approximately the same performance, which shows the strength of the developed approach in modeling and optimizing reliability. The performance of the optimal task reallocation strategies was evaluated and it was noticed that the service reliability was improved up to 65% as compared to the reliability provided by a DCS, and up to 22% as compared to policies that considered nodes' reliability but disregarded the communication costs over the network. Moreover, the algorithm developed in this work to devise task reallocation strategies achieved values for service reliability within 70% of the optimal service reliability, and in cases achieved the optimal value.

As a result of this work, fundamental tradeoffs and interplays between the different parameters governing the dynamics of DCSs were identified. First, due to limitations in the communication infrastructure, there is a tradeoff between delaying a reallocation action (in order to have an accurate account of the working or failed state of nodes) and immediately execute the reallocation strategy (to avoid wasting valuable computing time). The mathematical characterization for the reliability developed during this effort enabled us to optimally select when the reallocation action should be taken. Second, it was discovered that there is an interplay between the task transfer time and the idle time of the nodes; it was found that the service reliability can be improved if the idle times of the nodes are reduced as much as possible. Third, it was also found that effective task reallocation policies must consider the reliability of the nodes as a parameter. Fourth, it was found also that it is mandatory to consider the task-transfer delay when designing task reallocation strategies. When task-transfer delays are relatively larger than the execution time of tasks at the nodes, task

reallocation policies cannot be effective due to the excessive transfer-time taken by the tasks in the network. Finally, it was found that the model for the service reliability yields accurate predictions in operational conditions where the average task transfer times are less or approximately equal to the average service time of tasks.

2. Probabilistic Network Modeling: The Network Functionality

Executive Summary

We have completed the initial phase of construction of a mathematically rigorous formalism to describe the time evolution of the performance of any network subject to multiple random-in-space-and-time disruption and restoration events. Our main tool for quantifying network performance is the notion of *functionality*, which is roughly the ratio of (software) task execution time when the network is in an unimpaired state [numerator] to task execution time (for the identical task) when the network is impaired to some variable degree [denominator]. The initial phase quantifies the time evolution of the functionalities of individual network components (nodes and links); the results of the initial phase will then be used in a second phase in which the individual component behaviors are coalesced to describe the time evolution of the functionality of the network in its entirety. This model is applicable to network attack in general, and to WMD network attack in particular.

Publication

- D. Dietz and M. M. Hayat, "A Model for the Time Evolution of Networks Subject to Random Multiple Disruption and Restoration Events," in preparation.

Technical Summary

Introduction: We have completed the initial phase of construction of a mathematically rigorous formalism to describe the time evolution of the functionality of a network whose components are subject to multiple random-in-space-and-time impairment-causing and subsequent functionality-restoring events. Many situations may be envisaged for which such a model is applicable. One such situation of particular recent high interest is the potential attempt by terrorists to impair or destroy networks, via a WMD attack or otherwise, which form an integral part of civilian or military infrastructure. The objective of a terrorist attack on such a network is to cause it to function not at all or only partially for some period of time, ideally "forever" from the attackers' point of view. Nevertheless, it is to be expected that as time progresses subsequent to an attack, an affected network will be restored in stages to operability via restoration of the functionality of some, not necessarily all, of its impaired nodes and links ("components"). It is possible, however, that as restoration proceeds, additional attacks upon the network take place which serve to impair previously unaffected components as well as to re-impair some previously-impaired-but-restored components. Any network component, then, is subject in general to a finite time sequence of impairment and restoration events as the consequence of a finite time sequence of attacks; and the time evolution of the functionality of the network as a whole, being describable in terms of the collection of time evolutions of all of its individual components, is then also subject to such a sequence.

In order, therefore, to describe the time evolution of the functionality of a network subject to the conditions described above, we have constructed the initial portion–addressing only individual component behavior–of a model of network behavior which provides quantitative predictions of network (partial) functionality as each of the network components progresses in time through its own sequence of impairment and subsequent restoration events. Since the timing of the occurrences

of the impairment events and the amount of time required to subsequently restore the components are both stochastic, this model is also a stochastic one. This model consists of five basic concepts, namely (1) states, (2) functionality, (3) state and functionality time histories, (4) software task completion time, and (5) probability spaces for state and functionality time histories. We now discuss each of these briefly in turn in the context of a network consisting of C components (nodes and links) labeled by $\mu = 1, \ldots, C$.

Technical Details

States. A component labeled μ may, at any given instant, be in one of several possible attainable ***component states*** labeled by $\sigma_0, \sigma_1, \ldots, \sigma_{Q_\mu}$, where Q_μ is a positive integer, and we write $\Sigma^{(\mu)} \equiv \{\sigma_0^{(\mu)}, \ldots, \sigma_{Q_\mu}^{(\mu)}\}$ for the abstract set of all such states. The state $\sigma_0^{(\mu)}$ always represents the totally nonoperational condition of the component (i.e., being "dead") while the state $\sigma_{Q_\mu}^{(\mu)}$ always represents its totally operational condition (i.e., being capable of executing any appropriate assigned task at full performance capacity if so called upon); the states $\sigma_1, \ldots, \sigma_{Q_\mu - 1}$ allow for component intermediate levels of (partial) operationality between the two extremes and are labeled in no particular order. The network may then, at any given instant, be in one of several possible ***network states*** belonging to the set $\mathbf{\Sigma}^{[C]} \equiv X_{\mu=1}^{C} \Sigma^{(\mu)} = \{\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_{Q^{[C]}}\}$ of all attainable network states, where $Q^{[C]} + 1 = (Q_1 + 1)(Q_2 + 1)\ldots(Q_C + 1)$; also, $\boldsymbol{\sigma}_0 = \langle \boldsymbol{\sigma}_0^{(1)}, \boldsymbol{\sigma}_0^{(2)}, \ldots, \boldsymbol{\sigma}_0^{(C)} \rangle$ (the totally nonfunctional network state) and $\boldsymbol{\sigma}_{Q^{[C]}} = \langle \boldsymbol{\sigma}_{Q_1}^{(1)}, \boldsymbol{\sigma}_{Q_2}^{(2)}, \ldots, \boldsymbol{\sigma}_{Q_C}^{(C)} \rangle$ (the totally functional network state). **Functionality**. The ***functionality***, $\mathcal{F}$, of a component or a network, given a software task to be executed upon it, is defined notionally as the ratio

$$\mathcal{F} = \frac{\textit{Network} \text{ task completion time under } \textit{unimpaired} \text{ conditions}}{\textit{Network} \text{ task completion time under } \textit{impaired} \text{ conditions}}. \tag{4}$$

Functionality is task dependent and in general $0 \leq \mathcal{F} \leq 1$; for example, $\mathcal{F} = 1/2$ corresponds to a task execution slowdown by a factor of two. Further, for a component task, labeled say by Υ, we assign a numerical task dependent component state functionality $\phi_j(\Upsilon)$ to each of the states σ_j of $\Sigma^{[C]}$ via the ***component state functionality mapping*** $\mathcal{F}_\Upsilon^{\Sigma^{[C]}} : \Sigma^{[C]} \to [0,1]$ according to $\mathcal{F}_\Upsilon^{\Sigma^{[C]}}(\sigma_j) = \phi_j(\Upsilon), j = 0, \ldots, Q^{[C]}$, where $\mathcal{F}_\Upsilon^{\Sigma^{[C]}}(\sigma_0) = 0 = \phi_0$ and $\mathcal{F}_\Upsilon^{\Sigma^{[C]}}(\sigma_{Q^{[C]}}) = 1 = \phi_{Q^{[C]}}$. We denote the set of numerical component state functionalities by $\Phi_\Upsilon^{\Sigma^{[C]}} \subset [0,1]$. The set $\{\phi_j(\Upsilon)\}_{j=0}^{Q^{[C]}}$ of state functionality values must be supplied externally to the model.

Similarly, for a network task labeled by $\mathbf{\Upsilon}$, we will assign a numerical task dependent network state functionality $\boldsymbol{\phi}_j(\mathbf{\Upsilon})$ to each of the states $\boldsymbol{\sigma}_j$ of $\mathbf{\Sigma}^{[C]}$ via the ***network state functionality mapping*** $\boldsymbol{\mathcal{F}}_{\mathbf{\Upsilon}}^{\mathbf{\Sigma}^{[C]}} : \mathbf{\Sigma}^{[C]} \to [0,1]$ according to $\boldsymbol{\mathcal{F}}_{\mathbf{\Upsilon}}^{\mathbf{\Sigma}^{[C]}}(\boldsymbol{\sigma}_j) = \boldsymbol{\phi}_j(\mathbf{\Upsilon}), j = 0, \ldots, Q^{[C]}$, where $\boldsymbol{\mathcal{F}}_{\mathbf{\Upsilon}}^{\mathbf{\Sigma}^{[C]}}(\boldsymbol{\sigma}_0) = 0 = \boldsymbol{\phi}_0$ and $\boldsymbol{\mathcal{F}}_{\mathbf{\Upsilon}}^{\mathbf{\Sigma}^{[C]}}(\boldsymbol{\sigma}_{Q^{[C]}}) = 1 = \boldsymbol{\phi}_{Q^{[C]}}$. We will denote the set of numerical network state functionalities by $\mathbf{\Phi}_{\mathbf{\Upsilon}}^{\mathbf{\Sigma}^{[C]}} \subset [0,1]$. The set $\{\boldsymbol{\phi}_j(\mathbf{\Upsilon})\}_{j=0}^{Q^{[C]}}$ of state functionality values must be supplied externally to the model (and implicitly incorporate the network topology). These values may be computed in general by using a network performance code (e.g. OPNET).

Time Histories. In general, component μ proceeds in time through a sequence of states from $\Sigma^{(\mu)}$ as determined by the sequence of impairment and restoration events that it experiences. We represent this time evolution of the component instantaneous state by a function $h^{(\mu)} : [0, \infty) \to \Sigma^{(\mu)}$, termed a *component instantaneous state time history*, and defined by $h^{(\mu)}(t) = \sigma_j^{(\mu)}$ if the component is in state $\sigma_j^{(\mu)}$ at time instant t, $j \in \{0, \ldots, Q_\mu\}$. The function $f^{(h,\Upsilon)} \equiv \mathcal{F}_\Upsilon^{\Sigma^{[C]}} \circ h : [0, \infty) \to \Phi_\Upsilon^{\Sigma^{[C]}}$, termed a component instantaneous functionality time history, represents the time evolution of component instantaneous functionality when the component's state time history is $\mathbf{h}$ and the component task is Υ.

Similarly, a network proceeds in time through a sequence of states of $\mathbf{\Sigma}^{(\mu)}$ as determined by the sequences of impairment and restoration events that its components experience. This time evolution of the network instantaneous state will be represented by a function $\boldsymbol{h}$: $[0,\infty) \rightarrow \mathbf{\Sigma}^{(\mu)}$, termed a ***network instantaneous state time history***, defined by $\boldsymbol{h}(t) = \langle \boldsymbol{h}^{(0)}(t), \ldots, \boldsymbol{h}^{(C)}(t) \rangle$; indeed, $\boldsymbol{h}(t) = \boldsymbol{\sigma}_j$ if the network is in state $\boldsymbol{\sigma}_j$ at time t, $j \in \{0, \ldots, Q^{[C]}\}$. The function $\boldsymbol{f}^{(\boldsymbol{h},\mathbf{\Upsilon})} \equiv \boldsymbol{\mathcal{F}}_{\mathbf{\Upsilon}}^{\mathbf{\Sigma}^{[C]}} \circ \boldsymbol{h} : [0,\infty) \rightarrow \mathbf{\Phi}_{\mathbf{\Upsilon}}^{\mathbf{\Sigma}^{[C]}}$, termed a ***network instantaneous functionality time history***, will represent the time evolution of network instantaneous functionality when the network's state time history is $\boldsymbol{h}$ and the network task is $\mathbf{\Upsilon}$.

Task Completion Time. We define the component history task completion time for a component software task, Υ, running on a single computational network node (links do not participate here), which node is running only that task, and which node, in addition, has associated histories h and $f^{(h,\Upsilon)}$. Let Δ_0^{Υ} 0 denote the length of the time interval (duration) required for the entire task, Υ, to run from start to completion on the node if the node is forever unimpaired–always in state $\sigma_{Q^{[C]}}$–thus having state history $h \equiv \sigma_{Q^{[C]}}$ (i.e., $h^{(\mu)}(t) = \sigma_{Q_\mu}^{(\mu)}$ for all $t \geq 0$) and state functionality history $f^{(h,\Upsilon)} \equiv 1$. Δ_0^{Υ} 0 is assumed to be time translation invariant–independent of task starting instant, denoted t_0–on the forever-unimpaired node. In our model, any network task, Υ, being executed on the node is completely characterized by its Δ_0^{Υ} (plus its $\mathcal{F}_{\Upsilon}^{\Sigma^{[C]}}$) on the node. The ***component history task completion time*** (duration) for the node having time histories h and $f^{(h,\Upsilon)}$ is then defined by

$$\mathcal{C}_{[\Delta_0^{\Upsilon};\, t_0]}(f^{(h^{(\mu)},\Upsilon)}) \equiv \inf\{\tau \geq 0 \mid \mathcal{T}^{eff}_{(h^{(\mu)}\Upsilon)}(t_0;\tau) = \Delta_0^{\Upsilon}\} \tag{5}$$

where

$$\mathcal{T}^{eff}_{(h^{(\mu)},\Upsilon)}(t_0;\tau) \equiv \int_{t_0}^{t_0+\tau} f^{(h^{(\mu)},\Upsilon)}(t)dt. \tag{6}$$

The ***mean functionality***, taken over the entire task execution time interval, of node instantaneous functionality history $f^{(h,\Upsilon)}$ is

$$\mathcal{F}_{[\Delta_0^{\Upsilon};\, t_0]}(f^{(h^{(\mu)},\Upsilon)}) = [1/\mathcal{C}_{[\Delta_0^{\Upsilon};t_0]}(f^{(h^{(\mu)},\Upsilon)})] \int_{t_0}^{t_0+\mathcal{C}} f^{(h^{(\mu)},\Upsilon)}(t)dt = \Delta_0^{\Upsilon}/\mathcal{C}_{[\Delta_0^{\Upsilon};\, t_0]}(f^{(h^{(\mu)},\Upsilon)}), \tag{7}$$

where we take $\mathcal{F}_{[\Delta_0^{\Upsilon};t_0]}(f^{(h^{(\mu)},\Upsilon)}) = 0$ whenever $\mathcal{C}_{[\Delta_0^{\Upsilon};t_0]}(f^{(h^{(\mu)},\Upsilon)}) = \infty$. This yields precisely the same value as that of functionality $\mathcal{F}$ defined by the ratio in words given earlier.

Similarly, we will define the *network history task completion time* for a network software task, $\mathbf{\Upsilon}$, running entirely on the network, which network is running only that task (or collection of tasks) and which network, in addition, has associated histories $\boldsymbol{h}$ and $\boldsymbol{f}^{(\boldsymbol{h},\mathbf{\Upsilon})}$. Also, we will let $\boldsymbol{\Delta}_0^{\mathbf{\Upsilon}} > 0$ denote the length of the time interval (duration) required for the entire task, $\mathbf{\Upsilon}$, to run from start to completion on the network if the network is forever unimpaired–always in state $\boldsymbol{\sigma}_{Q^{[C]}}$–thus having state history $\boldsymbol{h} = \boldsymbol{\sigma}_{Q^{[C]}}$ and state functionality history $\boldsymbol{f}^{(\boldsymbol{h},\mathbf{\Upsilon})} \equiv 1$. $\boldsymbol{\Delta}_0^{\mathbf{\Upsilon}}$ is assumed to be time translation invariant–independent of task starting instant, denoted t_0 on the forever-unimpaired network. In our network model, any network task, $\mathbf{\Upsilon}$, being executed on the network will be completely characterized by its $\boldsymbol{\Delta}_0^{\mathbf{\Upsilon}}$ (plus its $\boldsymbol{\mathcal{F}}_{\mathbf{\Upsilon}}^{\mathbf{\Sigma}^{[C]}}$). The ***network history task completion time*** (duration) for the network having time histories $\boldsymbol{h}$ and $\boldsymbol{f}^{(\boldsymbol{h},\mathbf{\Upsilon})}$ and task starting instant $t_0 \geq 0$ will be defined by

$$\boldsymbol{\mathcal{C}}_{[\boldsymbol{\Delta}_0^{\mathbf{\Upsilon}};\, t_0]}(\boldsymbol{f}^{(\boldsymbol{h},\mathbf{\Upsilon})}) \equiv \inf\{\tau \geq 0 | \boldsymbol{\mathcal{T}}^{eff}_{(\boldsymbol{h},\mathbf{\Upsilon})}(t_0;\tau) = \boldsymbol{\Delta}_0^{\mathbf{\Upsilon}}\}$$

where

$$\boldsymbol{\mathcal{T}}^{eff}_{(\boldsymbol{h},\mathbf{\Upsilon})}(t_0;\tau) \equiv \int_{t_0}^{t_0+\tau} \boldsymbol{f}^{(\boldsymbol{h},\mathbf{\Upsilon})}(t)dt\ .$$

The *mean* functionality, taken over the entire task execution time interval, of network instantaneous functionality history $\boldsymbol{f}^{(\boldsymbol{h},\boldsymbol{\Upsilon})}$) will be

$$\boldsymbol{\mathcal{F}}_{[\boldsymbol{\Delta}_0^{\boldsymbol{\Upsilon}};\, t_0]}(\boldsymbol{f}^{(\boldsymbol{h},\boldsymbol{\Upsilon})}) = [1/\boldsymbol{\mathcal{C}}_{[\boldsymbol{\Delta}_0^{\boldsymbol{\Upsilon}};t_0]}(\boldsymbol{f}^{(\boldsymbol{h},\boldsymbol{\Upsilon})})] \int_{t_0}^{t_0+\boldsymbol{\mathcal{C}}} \boldsymbol{f}^{(\boldsymbol{h},\boldsymbol{\Upsilon})}(t)dt = \boldsymbol{\Delta}_0^{\boldsymbol{\Upsilon}}/\boldsymbol{\mathcal{C}}_{[\boldsymbol{\Delta}_0^{\boldsymbol{\Upsilon}};\, t_0]}(\boldsymbol{f}^{(\boldsymbol{h},\boldsymbol{\Upsilon})}),$$

where we take $\boldsymbol{\mathcal{F}}_{[\boldsymbol{\Delta}_0^{\boldsymbol{\Upsilon}};t_0]}(\boldsymbol{f}^{(\boldsymbol{h},\boldsymbol{\Upsilon})}) = 0$ whenever $\boldsymbol{\mathcal{C}}_{[\boldsymbol{\Delta}_0^{\boldsymbol{\Upsilon}};t_0]}(\boldsymbol{f}^{(\boldsymbol{h},\boldsymbol{\Upsilon})}) = \infty$. This yields precisely the same value as that of functionality $\mathcal{F}$ defined by the ratio in words given earlier.

Probability Spaces. As was pointed out in the Introduction, both the timing of the occurrences of impairment events in a network, via impairment of its components, and the amount of time required to (partially) restore the network, via restoration of impaired components, are stochastic. In other words, each possible (state or functionality) history that a given component, hence network, may experience as a result of a sequence of impairments and subsequent restorations must be treated as a member of an appropriate probability space. Three families of such probability spaces are required in our formalism: one family, $\{(\mathcal{H}^{\Sigma^{(\mu)}}, \mathcal{A}^{\Sigma^{(\mu)}}, \mathcal{P}^{\Sigma^{(\mu)}})\}_{\mu=1}^{C}$, of probability spaces for component state histories associated with the collection of all the components which the network comprises; a second, $(\boldsymbol{\mathcal{H}}^{\boldsymbol{\Sigma}}, \boldsymbol{\mathcal{A}}^{\boldsymbol{\Sigma}}, \boldsymbol{\mathcal{P}}^{\boldsymbol{\Sigma}})$, for network state histories; and a third family, $\{(\boldsymbol{\mathcal{H}}^{\boldsymbol{\Phi}_{\boldsymbol{\Upsilon}}}, \boldsymbol{\mathcal{A}}^{\boldsymbol{\Phi}_{\boldsymbol{\Upsilon}}}, \boldsymbol{\mathcal{P}}^{\boldsymbol{\Phi}_{\boldsymbol{\Upsilon}}})\}_{\boldsymbol{\Upsilon}}$, for network functionality histories indexed by task label $\boldsymbol{\Upsilon}$. The third family is in fact induced by the second which is in turn induced by the first. We have constructed the first family under the current effort; the second and third families will be constructed under a follow-on effort. We point out that the probability measures $\mathcal{P}^{\Sigma^{(\mu)}}$ are perfectly arbitrary and are to be inferred from the conditions imposed by any given network attack/restoration scenario. Also, in this setting, $\mathcal{C}_{[\Delta_0^{\Upsilon};t_0]}$ and $\mathcal{F}_{[\Delta_0^{\Upsilon};t_0]}$ are both random variables. We may then compute distributions and expectations (as well as higher moments) of these random variables with respect to probability assignment $\mathcal{P}^{\Phi_{\Upsilon}}$ to quantify partially-impaired/partially-restored component effectiveness vis-a-vis any component task. Furthermore, in this setting, $\boldsymbol{\mathcal{C}}_{[\boldsymbol{\Delta}_0^{\boldsymbol{\Upsilon}};t_0]}$ and $\boldsymbol{\mathcal{F}}_{[\boldsymbol{\Delta}_0^{\boldsymbol{\Upsilon}};t_0]}$ are both to be random variables in $(\boldsymbol{\mathcal{H}}^{\boldsymbol{\Phi}_{\boldsymbol{\Upsilon}}}, \boldsymbol{\mathcal{A}}^{\boldsymbol{\Phi}_{\boldsymbol{\Upsilon}}}, \boldsymbol{\mathcal{P}}^{\boldsymbol{\Phi}_{\boldsymbol{\Upsilon}}})$; we will then be able to compute distributions of these with respect to probability assignment $\boldsymbol{\mathcal{P}}^{\boldsymbol{\Phi}_{\boldsymbol{\Upsilon}}}$ to quantify partially-impaired/partially-restored network effectiveness.

3. Network Health Assessment

Executive Summary

In this part, we consider a network that is trying to reach consensus on the occurrence of a WMD attack, by communicating over Additive White Gaussian Noise (AWGN) channels. We develop the mathematical foundations of networked binary consensus over noisy links. We first consider the case where the nodes do not have any knowledge of link qualities. We show that the asymptotic behavior in the presence of any amount of non-zero communication noise becomes unfavorable as the network loses the memory of the initial state. However, we show that the network can still reach and stay in accurate consensus for a long period of time. In order to characterize this, we derive a tight approximation for the second largest eigenvalue of the network and show how it is related to the size of the network and communication noise variance. We then consider the case where knowledge of the corresponding link qualities is available at every receiving node. We extend our framework and propose novel soft information processing approaches to improve the performance in the presence of noisy links. We show that, by learning the voting patterns, we can solve the undesirable asymptotic behavior of binary consensus. We furthermore characterize the impact of network connectivity on consensus performance. Finally, we show the underlying tradeoffs between robustness to link error and optimization of information flow that arise in networked binary consensus over noisy links.

Our achievements are motivated by and directly related to a network that is under a WMD attack. In case of such an attack, the knowledge available for proper, timely and accurate detection

of it is limited, sparse and prone to errors. Then distributed assessment of network health in the presence of uncertainties becomes considerably important, which is the main motivation for our developed framework. While there exists several work on estimation consensus problems, detection consensus, as relevant to WMD attacks, has not received much attention in the literature. Our proposed framework has therefore pushed the boundaries of what was known in the literature.

Personnel Supported

The following personnel worked on problems related to this portion of the grant:

- Graduate Students: Alejandro Gonzalez Ruiz (fullbright scholar), Yongxian Ruan
- Faculty: Yasamin Mostofi (Co-PI)

Publications

- Y. Mostofi, "Binary Consensus with Gaussian Communication Noise: A Probabilistic Approach," Proceedings of the 46th IEEE Conference on Decision and Control (CDC), Dec. 2007
- Y. Ruan and Y. Mostofi, "Binary Consensus with Soft Information Processing in Cooperative Networks," invited paper, 47th IEEE Conference on Decision and Control (CDC), 2008.
- Y. Yuan and Y. Mostofi, "Impact of Link Qualities and Network Topology on Binary Consensus," American Control Conference (ACC), 2009
- M. Malmirchegini, Y. Ruan and Y. Mostofi, "Binary Consensus Over Fading Channels: A Best Affine Estimation Approach," IEEE Globecom, 2008.
- A. Gonzalez Ruiz and Y. Mostofi, "Distributed Load Balancing over Directed Network Topologies," best paper in the session, American Control Conference (ACC), 2009.

Technical Summary

Early and accurate detection of the presence of WMD stressors is crucial to robust recovery of the network. The knowledge available for such detection, however, is limited and sparse. Each node will make a detection decision based on the information that is available to it. The knowledge available at each node, however, is limited and could be corrupted if part of the network is already compromised by the attack. Furthermore, link qualities can be far from ideal due to natural phenomena such as noise or as a result of a WMD attack. Therefore, the network should rely on collective information processing to make a more accurate decision on the detection of WMD stressors. Consensus problems arise when a group of distributed nodes need to reach an agreement on the value of a parameter of interest and can be categorized into two main groups: Estimation Consensus and Detection Consensus. Estimation consensus refers to the problems where the parameter of interest can take values over an infinite set or an unknown finite set. In general, such problems have received considerable attention in the literature [9, 10] (except for considering the impact of uncertainties on such problems, which have received lesser attention). By detection consensus, on the other hand, we refer to the problems in which the parameter of interest takes values from a finite known set. Then the update protocol that each agent will utilize becomes non-linear. We referred to a subset of detection consensus problems where the network is trying to reach an agreement over a parameter that can only have two values as binary consensus [5]. For instance, networked detection of a WMD attack falls into this category. While there exists a rich literature on estimation consensus, detection consensus problems only recently started to receive attention,

with our work being one of the early ones in this area. In this part, we discuss our main results along this line.

Consider M agents that want to reach consensus on the occurrence of a WMD attack. Each agent makes a decision on the occurrence of the event based on its one-time local sensor measurement. Let $b_i(0) \in \{0,1\}$ represent the initial decision of the i^{th} agent, at time step $k=0$, based on its local measurement. $b_i = 1$ indicates that the i^{th} agent votes that the event occurred whereas $b_i = 0$ denotes otherwise. Each agent sends its vote to its neighbors, using only one bit of information, and revises its vote based on the received information. Each transmission gets corrupted by the receiver noise, which is best modeled by Additive White Gaussian Noise (AWGN channel). Let $w_{j,i}(k)$ represent the noise at the k^{th} time step in the transmission of the information from the j^{th} node to the i^{th} one. $w_{j,i}(k)$ is a zero-mean Gaussian random variable with the variance of σ^2.

In this research effort, we consider a connected undirected time-invariant graph. Consider the case where the j^{th} agent can communicate (albeit noisy) to the i^{th} one. Let random variable $b_{j,i}(k)$ represent the reception of the i^{th} agent from the transmission of the j^{th} one at the k^{th} time step for $k \in \mathbb{N}_0$. We will have

$$b_{j,i}(k) = b_j(k) + w_{j,i}(k), \text{ for } j \in \Psi_i, \tag{8}$$

where Ψ_i represents the set of those agents that can communicate to the i^{th} one (including itself) and $w_{i,i} = 0$. We have

$$\Psi_i = \{z_i(1), z_i(2), \ldots, z_i(N_i)\} \text{ for } z_i(j) \in \{1, 2, \ldots, M\}, \tag{9}$$

where $z_i(j) \neq z_i(j')$ for $j \neq j'$, $1 \leq j, j' \leq N_i$, and $1 < N_i \leq M$ represents the size of Ψ_i. Each agent will then update its vote based on the received information as follows:

$$b_i(k+1) = F\left(b_{z_i(1),i}(k), b_{z_i(2),i}(k), \ldots, b_{z_i(N_i),i}(k)\right), \text{ for } z_i(j) \in \Psi_i \text{ and } 1 \leq j \leq N_i, \tag{10}$$

where $b_{i,i} = b_i$ and $F(.)$ represents a decision-making function. It should be noted that $b_i(k+1)$ is a random variable through its dependency on the reception noise. As part of this research effort, we consider different ways of building the decision-making function based on the knowledge available on link qualities. Let $\Omega_M = \{0, 1, 2, \ldots, M\}$ represent a finite set and

$$S(k) = \sum_{j=1}^{M} b_j(k) \in \Omega_M. \tag{11}$$

$S(k)$ is a measure of the closeness to consensus at time step k.

Definition: We say that the network is in an *accurate* consensus state at the k^{th} time step if and only if the following holds:

$$\begin{aligned} &\text{if} \quad S(0) \geq \lceil \frac{M}{2} \rceil \Rightarrow \forall i \quad b_i(k) = 1, \\ &\text{if} \quad S(0) < \lceil \frac{M}{2} \rceil \Rightarrow \forall i \quad b_i(k) = 0. \end{aligned} \tag{12}$$

Due to the presence of Gaussian noise, there is no guarantee in reaching or staying in consensus. In other words, there is no transition point beyond which consensus is reached permanently. Rather than that, we are interested in evaluating the probability of reaching and staying in different states of the network, asymptotic probabilistic behavior of the system as well as relating these parameters to the noise variance, size of the network and network connectivity.

3.1 Binary Consensus over a Fully-Connected Graph – Case of Unknown Link Quality

In order to characterize the impact of noisy links on binary consensus of a WMD attack, we first consider consensus over a fully-connected network, where there is a link, albeit noisy, from every node to every other node in the network. Under full-connectivity assumption, we will have the following for $k \geq 0$,

$$\begin{aligned} b_i(k+1) &= \text{Dec}\left(\frac{1}{M}\sum_{j=1}^{M} b_{j,i}(k)\right) \\ &= \text{Dec}\left(\frac{S(k)}{M} + \frac{1}{M}\sum_{j=1, j\neq i}^{M} w_{j,i}(k)\right), \end{aligned} \tag{13}$$

where Dec(.) represents a decision function for binary 0-1 detection: $\text{Dec}(x) = \begin{cases} 1 & x \geq .5 \\ 0 & x < .5 \end{cases}$.

Let $\Pi_n(k)$ represent the probability that $n \in \Omega_M$ of the agents are voting 1 at the k^{th} time step:

$$\Pi_n(k) = \text{Prob}\left[S(k) = n\right], \quad n \in \Omega_M, k \geq 0. \tag{14}$$

Let $P_{n,m}$ represent the probability of the network going from state n to state m in one time step,

$$P_{n,m} = \text{Prob}\left[S(k+1) = m | S(k) = n\right], \quad n, m \in \Omega_M. \tag{15}$$

More specifically, $P_{n,m}$ will have a binomial distribution as follows,

$$P_{n,m} = \binom{M}{m} \kappa_{n,M}^{m} (1 - \kappa_{n,M})^{M-m}, \tag{16}$$

where $\kappa_{n,M}$ represents the probability that any agent votes 1 in the next time step, given a current state of n ($S(k) = n$). This probability is the same for all the agents. Consider the i^{th} agent. We will have:

$$\kappa_{n,M} = \text{Prob}\left[\frac{n}{M} + W_i(k) > .5\right] = Q\left(\frac{.5 - \frac{n}{M}}{\sigma_M}\right), \tag{17}$$

where $W_i(k) = \frac{1}{M}\sum_{j=1}^{M} w_{j,i}(k)$ has a Gaussian distribution with zero mean and variance of $\sigma_M^2 = \frac{(M-1)\sigma^2}{M^2}$ and

$$\Pi(k+1) = P^T \Pi(k). \tag{18}$$

Remark 1. *It can be easily confirmed, using Gersgorin disk theorem [3], that the eigenvalues of P are located in the following area: $\bigcup_{n=0}^{M}\{z \in C : |z - P_{n,n}| \leq 1 - P_{n,n}\}$. Let $\lambda_{0,fc}, \lambda_{1,fc}, \ldots, \lambda_{M,fc}$ represent the eigenvalues of matrix P in a decreasing order: $|\lambda_{0,fc}| \geq |\lambda_{1,fc}| \geq \ldots \geq |\lambda_{M,fc}|$. This implies that $\forall n, |\lambda_{n,fc}| \leq 1$.*

Remark 2. *Assume $\sigma \neq 0$. Then $P > 0$ (element-wise). From stochastic property of matrix P, we know that one is an eigenvalue. From Remark 1 and applying Perron's theorem [3], we will have,*
a) $\lambda_{0,fc} = 1$ as a simple eigenvalue of P,
b) $|\lambda_{n,fc}| < 1$ for $n \neq 0$ and
c) $\left[P^T\right]^k \to L$ as $k \to \infty$, where $L = ZZ_{\text{left}}^T$, $Z = P^T Z$, $Z_{\text{left}} = PZ_{\text{left}}$, and $Z^T Z_{\text{left}} = 1$.

Lemma 1. *Matrix P is centro-symmetric, i.e. the $(M-n)^{th}$ row (or column) of matrix P is a reverse repeated version of the n^{th} row (or column) for $n \in \Omega_M$: $P_{n,m} = P_{M-n,M-m}$ for $n, m \in \Omega_M$.*

Proof: See our proof in [5].

If $\sigma \neq 0$, from Remark 2 we know that the network reaches a steady state asymptotically. Furthermore, we will have $\lim_{k\to\infty} \Pi(k) = Z Z_{\text{left}}^T \Pi(0)$ where Z and Z_{left} are as defined in Remark 2.

Remark 3. *Consider $Z Z_{\text{left}}^T \Pi(0)$, the asymptotic value of vector Π. $\Pi(0)$ has exactly one element equal to one and the rest zero while Z_{left} is a vector whose elements are all the same. Then, $Z_{\text{left}}^T \Pi(0)$ loses the information of the initial state. Therefore, the asymptotic value will be independent of the initial state.*

Remark 3 shows that, asymptotically, the information of the initial state will be lost, in the presence of any amount of non-zero communication noise. However, the system can still reach and stay in accurate consensus for a long period of time (which could be enough for practical purposes). Fig. 2 shows $\Pi(k)$ as a function of time step, k, for $M = 4$ agents. Initially, 3 out of 4 nodes vote one initially. Then the system is in accurate consensus at time step k if all nodes are voting one. The solid line shows the probability of having an accurate consensus ($\Pi_4(k)$). It can be seen that for a good amount of time (enough for practical purposes), the system will be in accurate consensus with high probability. It should be noted that the link noise is rather high for this example. To see the impact of link quality, Fig. 3 shows the impact of link quality on binary consensus over a fully-connected graph of 4 nodes, where 3 out of 4 vote 1 initially. In general, the smaller the communication noise variance is, the higher the probability of reaching and maintaining an accurate consensus will be, as can be seen from Fig. 3. The second largest eigenvalue ($\lambda_{1,\text{fc}}$) plays a key role in determining how fast the network is approaching its steady state. The closer the second eigenvalue is to the unit circle, the slower the rate of convergence, which results in consensus for a longer period of time. We will next derive an expression for the second largest eigenvalue of matrix P.

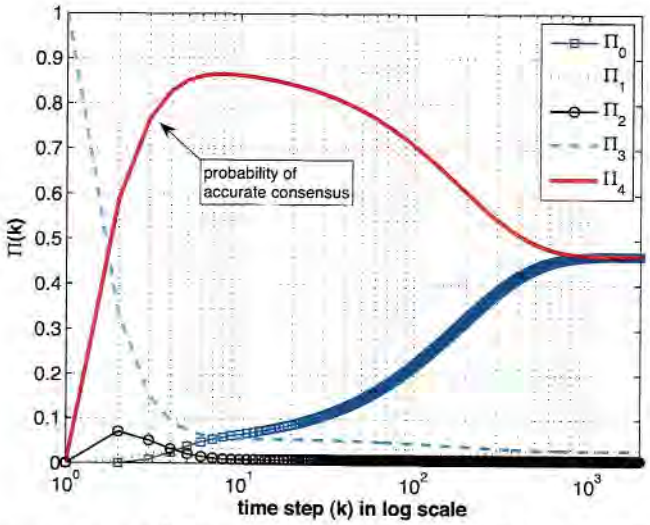


Figure 2. Consensus dynamics over a fully-connected graph – case of unknown link quality, $M = 4$, $\sigma = 0.5$.

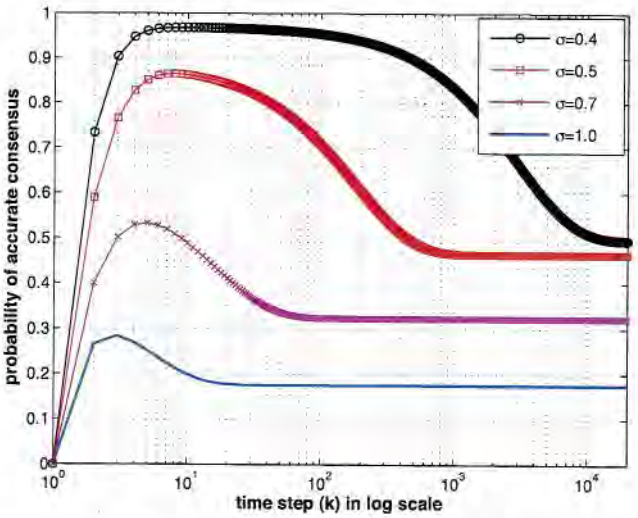


Figure 3. Impact of link quality on binary consensus – case of fully-connected graph with unknown link quality, $M = 4$.

Lemma 2. *Let P represent a transition probability matrix generated using $\kappa_{n,M}$, as indicated by Eq. 16. We will have,*

$$\sum_{m=0}^{\frac{M}{2}-1} \left(\frac{M}{2} - m\right)\left(P_{n,m} - P_{M-n,m}\right) = \frac{M}{2}(1 - 2\kappa_{n,M}). \tag{19}$$

Proof: See our proof in [5].

Using a number of other derived lemmas, we can then prove the following theorem regarding the second largest eigenvalue:

Theorem 1. *Let P_{approx} represent an approximation of matrix P under the linearization of Q function around the original. Let $\lambda_{1,fc,approx}$ represent the second largest eigenvalue of P_{approx}. Then we will have, $\lambda_{1,fc,approx} = 1 - 2\kappa_{0,M} = 1 - 2Q\left(\frac{1}{2\sigma_M}\right)$.*

Proof: See our proof in [5].

To see how well the approximation of Theorem 1 works, Fig. 4 shows the second largest eigenvalue and its approximation as a function of σ and for $M = 4$, $M = 7$ and $M = 16$. As can be seen, the approximation works well especially for smaller M and larger σ. The proposed approximation can be considerably useful in understanding the behavior of group consensus in terms of probability of reaching and staying in consensus.

3.2 Soft Information processing – Case of Known Link Quality

In this section, we consider the case where each node has the knowledge of its link qualities (variance of the noise). We propose novel information processing techniques for improving the performance of binary consensus, based on this knowledge. Then, we have the following decision-making function

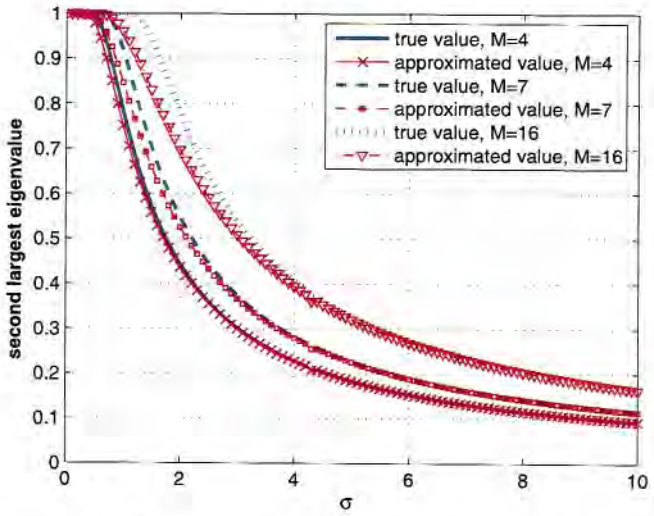


Figure 4. Approximation of the second largest eigenvalue.

at each node after linearization.

$$\begin{aligned} b_i(k+1) &= \text{Dec}\left(\frac{1}{M}\left[b_i(k)+\sum_{j=1,j\neq i}^{M}\Big(\alpha b_{j,i}(k)+\beta\Big)\right]\right) \\ &= \text{Dec}\left(\frac{1}{M}\left[b_i(k)+\alpha\sum_{j=1,j\neq i}^{M} b_j(k)\right]+\gamma+\alpha W_i(k)\right), \end{aligned} \tag{20}$$

where the last term $(\alpha W_i(k))$ is a zero-mean Gaussian noise with the variance of σ_s^2,

$$\alpha=\frac{1}{1+4\sigma^2}, \sigma_s^2=\frac{(M-1)\alpha^2\sigma^2}{M^2}, \text{ and } \gamma=\frac{M-1}{2M}(1-\alpha). \tag{21}$$

We can extend the framework of the previous section and define the following variables:

$$\begin{aligned} \kappa_{n,M|1} =& \text{Prob}[b_i(k+1)=1|b_i(k)=1 \text{ and } S(k)=n] \\ =& Q\left(\frac{0.5-\gamma-\frac{1+(n-1)\alpha}{M}}{\sigma_s}\right), 1\leq i\leq M, \\ \kappa_{n,M|0} =& \text{Prob}[b_i(k+1)=1|b_i(k)=0 \text{ and } S(k)=n] \\ =& Q\left(\frac{0.5-\gamma-\frac{n\alpha}{M}}{\sigma_s}\right), 1\leq i\leq M \end{aligned} \tag{22}$$

and

$$\begin{aligned} P_{\text{soft},n,m} =& \text{Prob}[S(k+1)=m|S(k)=n] \\ =& \sum_{x=\psi_{n,m}}^{\psi'_{n,m}} f(x,n,\kappa_{n,M|1})f(m-x,M-n,\kappa_{n,M|0}) \\ =& \sum_{x=\psi_{n,m}}^{\psi'_{n,m}} \binom{n}{x}\kappa_{n,M|1}^{x}(1-\kappa_{n,M|1})^{n-x}\times\binom{M-n}{m-x}\kappa_{n,M|0}^{m-x}(1-\kappa_{n,M|0})^{M-n-m+x}, \end{aligned} \tag{23}$$

where $\psi_{n,m} = \max(0, m+n-M)$, $\psi'_{n,m} = \min(n,m)$ and $f(x,n,q) = \binom{n}{x} q^x (1-q)^{n-x}$ is the pdf of a binomial distribution with q as the success probability. The rate of convergence of the binary consensus with soft information processing is characterized by calculating the second eigenvalue of the transition matrix.

Lemma 3. $\kappa_{n,M|1} = 1 - \kappa_{M-n,M|0}$.

Proof: see our proof in [11].

Lemma 4. *Matrix P_{soft} is a centrosymmetric matrix: $P_{soft,M-n,M-m} = P_{soft,n,m}$ for $0 \leq n, m \leq M$.*

Proof: see our proof in [11].

Theorem 2. *Let $P_{soft,approx}$ represent the transition probability matrix generated under the linearization of Q function around the original. Let $\lambda^{soft}_{1,fc,approx}$ represent the second largest eigenvalue of $P_{soft,approx}$. Then we have, $\lambda^{soft}_{1,fc,approx} = 1 - 2\eta_0 = 1 - 2Q(\frac{4\sigma^2+M}{2\sigma\sqrt{M-1}})$.*

Proof: see our proof in [11].

To see how well the approximation of Theorem 2 works, Fig. 5 shows the second largest eigenvalue and its approximation as a function of σ and for $M = 4$, $M = 10$ and $M = 20$. As can be seen, the approximation works well especially for smaller M, larger σ, or σ close to zero. For comparison, the second largest eigenvalue was proved to be $\lambda_{1,\text{fc,approx}} = 1 - 2Q(\frac{M}{2\sigma\sqrt{M-1}})$ in Theorem 1 for the case where no channel knowledge was available.

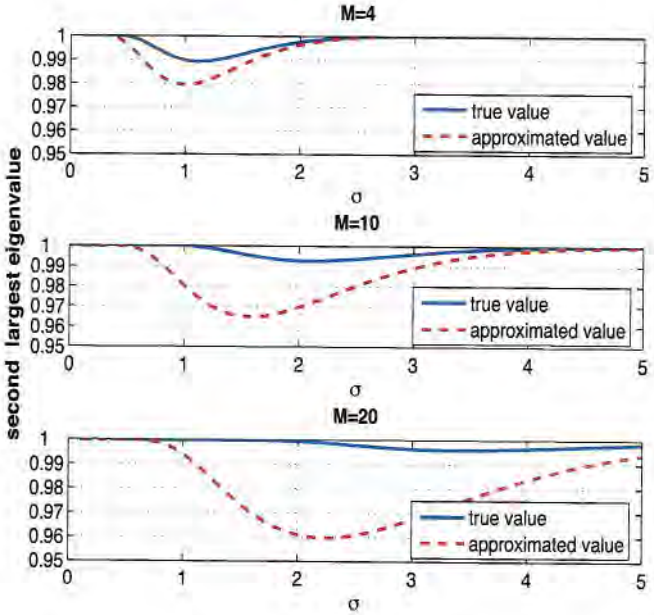


Figure 5. Approximation of the second largest eigenvalue for basic soft case.

3.2.1 Soft Information Processing with Learning

In the true structure of soft information processing, the i^{th} node needs to have the knowledge of $\text{Prob}[b_j(k) = 1]$ for $j \neq i$. Such information is not readily available in the receiver. However, it can be statistically estimated. Let $\hat{p}_{j,i}(k)$ represent the i^{th} node's estimate of $\text{Prob}[b_j(k) = 1]$. Then we will have the following form of decision-making:

$$b_i(k+1) = \text{Dec}\left(\frac{1}{M}\left[b_i(k) + \sum_{j\neq i} \frac{\hat{p}_{j,i}(k)}{\hat{p}_{j,i}(k) + (1-\hat{p}_{j,i}(k))\, e^{\frac{-2b_{j,i}(k)+1}{2\sigma^2}}}\right]\right). \tag{24}$$

Fig. 6 shows the performance of the proposed soft information processing approaches for a network of 4 nodes with $\sigma = 1$. $\sigma = 1$ corresponds to Signal to Noise Ratio of -3dB per link, which is extremely low. For comparison, the performance for the case where no channel knowledge is available is also plotted. The basic soft information processing (with no learning) can improve the performance of reaching consensus over the network. However, its asymptotic behavior is still undesirable as the probability of accurate consensus starts to decrease after a while. It can be seen that the proposed soft strategies can increase the probability of accurate consensus considerably. It can be seen that by estimating the probability distribution of the votes of other nodes, learning soft can improve the performance considerably and have a desirable asymptotic behavior (the system preserves the memory of the initial state) as well as a superior transient behavior.

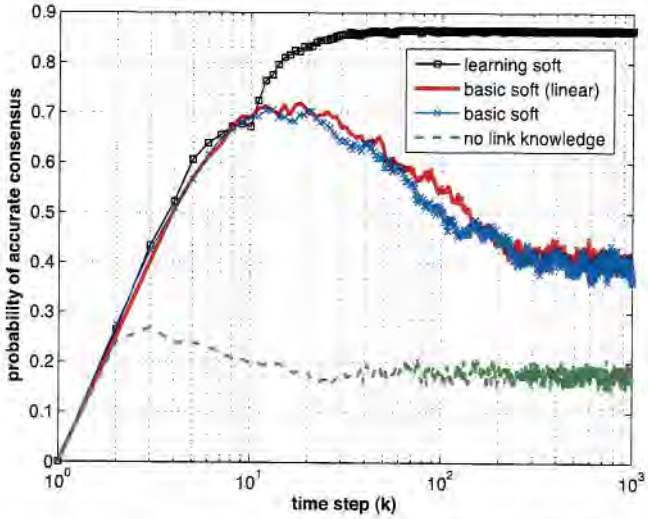


Figure 6. Performance of the proposed soft-information processing approaches for $M = 4$ and $\sigma = 1$.

3.3 Impact of Graph Connectivity

In this part we discuss our result on the impact of graphs that are not fully connected on the consensus behavior. More specifically, we extend our framework to derive an expression for the second largest eigenvalue assuming a time-invariant connected (undirected) graph.

3.3.1 Case of Unknown Link Quality

If the graph is not fully connected, we will have

$$\begin{aligned} b_i(k+1) &= \text{Dec}\left(\frac{1}{N_i}\sum_{j\in\Psi_i} b_{j,i}(k)\right) \\ &= \text{Dec}\left(\frac{1}{N_i}\sum_{j\in\Psi_i} b_j(k) + \frac{1}{N_i}\sum_{j\in\Psi_i, j\neq i} w_{j,i}(k)\right). \end{aligned} \tag{25}$$

where Ψ_i represents the set of those agents that can communicate to the i^{th} one (including itself) and N_i represents the size of Ψ_i.

Theorem 3. *Let T_{approx} represent the higher-order dimensional transition probability matrix generated under the linearization of Q function around the original. Let $N_i = N$ for $1 \leq i \leq M$. Then, $\lambda_{1,nfc,approx} = 1 - 2Q(\frac{1}{2\sigma_N})$ is the second largest eigenvalue of T_{approx}, where $\sigma_N^2 = \frac{(N-1)\sigma^2}{N^2}$ and subscript "nfc" denotes the eigenvalues for the case of not fully-connected graphs.*

Proof: see our proof in [11, 12].

3.3.2 Case of Known Link Quality

The analysis of basic soft-information processing can be extended to not fully-connected graphs as follows,

$$\begin{aligned} b_i(k+1) &= \text{Dec}\left(\frac{1}{N_i}\left[b_i(k) + \sum_{j\in\Psi_i, j\neq i}\Big(\alpha b_{j,i}(k) + \beta\Big)\right]\right) \\ &= \text{Dec}\left(\frac{1}{N_i}\left[b_i(k) + \alpha\sum_{j\in\Psi_i, j\neq i} b_j(k)\right] + \gamma_i + \alpha W_i(k)\right), \end{aligned} \tag{26}$$

where $W_i(k) = \sum_{j\in\Psi_i, j\neq i} w_{j,i}(k)$ and $\gamma_i = \frac{N_i-1}{2N_i}(1-\alpha)$.

Theorem 4. *Let $T_{soft,approx}$ represent the higher-order dimensional transition probability matrix generated under the linearization of Q function around the original. Let $N_i = N$ for $1 \leq i \leq M$. Then, $\lambda^{soft}_{1,nfc,approx} = 1 - 2\kappa_{0,N|0}$ is the second largest eigenvalue of $T_{soft,approx}$, where $\kappa_{0,N|0} = Q\left(\frac{\frac{1}{2}-\gamma_i}{\sigma_{s_i}}\right)$ for any $1 \leq i \leq M$, where $\sigma_{s_i}^2 = \frac{(N_i-1)\alpha^2\sigma^2}{N_i^2}$.*

Proof: see our proof in [6].

4. Active Data Management

Executive Summary

This portion of the research considered computation and storage in distributed, unreliable sensor networks subject to significant, potentially correlated failures induced by WMD stressors. To enable such systems, this research focused on lightweight monitoring systems that can drive decentralized adaptation. Primarily, we considered making optimistic adaptation decisions using previously-developed systems for lightweight gossiping of performance data on existing application messages [13, 14]. As part of this research, we developed a set of metrics to quantify the suitability

of an application to gossip-based monitoring and adaptation, developed a variant of the two-phase commit protocol needed for adaptation using gossiped performance data, and studied how effective gossip-based performance data was in driving load-balancing decisions in distributed applications. In addition, we examined system software extensions to allow us to quantify the impact of failures, including the correlated failures caused by WMD stressors on network processing. Preliminary results from this research demonstrate the ability to control fault injection in sensor systems both in simulation and real hardware, providing a viable testbed for performing basic networking research on WMD-induced correlated failures in sensor systems.

Personnel

The following personnel working on this portion of the research:

- Graduate Students: F. Orlando Arbildo, Ricardo Villalon
- Undergraduate Students: Basak Gocmen, UNM; Thomas Jones, Sean Dardis, and Tyler Halva, Gonzaga
- Postdoctoral Fellows: Patrick M. Widener, Wenbin Zhu
- Faculty: Patrick Bridges (PI), UNM; Patricia Crowley (Co-PI), Gonzaga University

Publications

- Wenbin Zhu, Patrick G. Bridges, and Arthur B. Maccabe, "Lightweight application monitoring and tuning with embedded gossip." IEEE Transactions of Parallel and Distributed Systems (TPDS), 20(7):1038-1049, July 2009.
- F. Orlando Arbildo and Basak Gocmen, "A Compact Testbed for Wireless Sensor Networks", UNM Computer Science Student Research Conference Poster Session, 2009.

Technical Summary

Quantifying Suitability to Gossip-based Monitoring: Driving adaptation in large-scale failure-prone distributed systems such as battlefield sensor network systems requires appropriate monitoring information. Recent research [14] has shown that such monitoring can be done scalably if performance data is gossiped on existing application messages with the cost of sacrificing the global consistency of performance data. In this portion of the research, we sought to quantify the usefulness of this gossip-based approach to various optimizations by developing an appropriate set of metrics.

We developed five different metrics for evaluating the suitability of a given application to gossip-based distributed monitoring and adaptation techniques. These metrics include three time-interval metrics that measure the largest amount of time it could take to perform a portion of gossip-based communication in an application, summarized in figure 7, as well as two others. Specifically, the five metrics we have chosen are:

- **Wait time,** the amount of time between taking a measurement and the first remote node receiving complete monitoring information.
- **Propagation time,** the amount of time between the first and the last remote nodes receiving complete monitoring information.
- **Resolution time,** the amount of time between taking a measurement and the last remote node receiving complete monitoring information (i.e., the sum of wait time and propagation time).

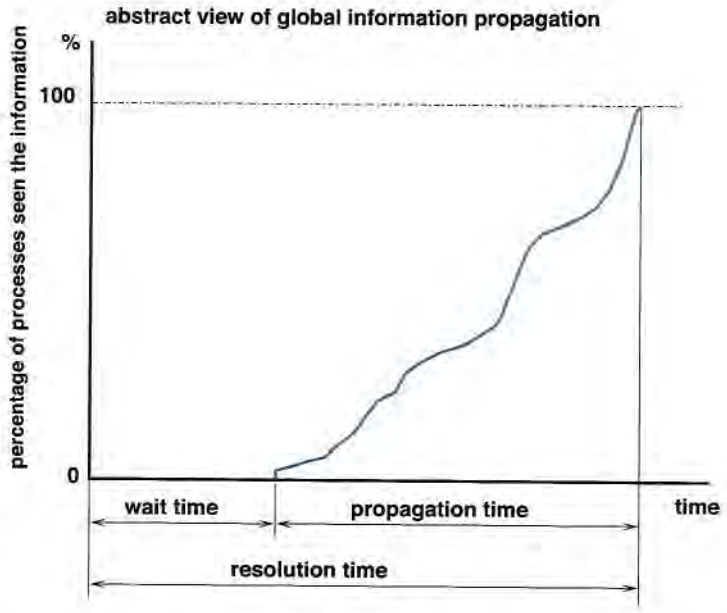


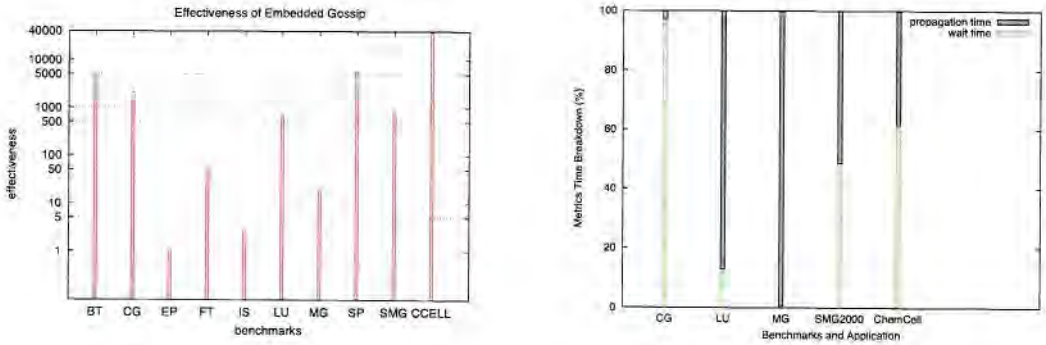
Figure 7. Time-interval metrics: wait time, propagation time, and resolution time are defined in terms of when the first and last remote nodes receive data measured at time $t = 0$.

- **Effectiveness,** the number of complete measurements that can be done by Embedded Gossip during an application's execution.
- **Monitoring overhead,** the cost of doing embedded gossiping in an application.

Figure 8(a) shows the effectiveness results of 9 benchmarks and one application. We can see that some benchmarks, *BT, CG, LU, SP, SMG,* and CCELL, can do more than 500 one-to-all notifications during their execution time. Based on this, the types of applications represented by each of these benchmarks, all common high-performance application types, appear to be amenable to gossip-based monitoring. In contrast, Embedded Gossip performs poorly for *EP* and *IS* because of the embarassingly parallel nature of these applications.

Figure 8(b) shows the timer interval metrics for four benchmarks and one application that all have high effectiveness metrics, *CG, LU, MG, SMG2000* and *ChemCell.* Despite having roughly similar effectiveness, the breakdown between wait time and propagation time in these applications varies dramatically. *CG,* for example, spends the majority of their resolution time waiting for a measurement to begin to propagate, and then spreads the resulting data quickly throughout the application, resulting in a relatively small amount of time during which different process's global views are inconsistent. Programs like *LU* or *MG,* on the other hand, gradually propagate results throughout the application, resulting in relatively longer propagation times than wait times. Because of this, gossip-based approaches are unlikely to be appropriate for driving global adaptation in programs structured like *LU* or *MG* despite being an effective measurement tool in these applications. Programs like *SMG2000* and *ChemCell* represent a middle ground between these two application, with both resolution time split relatively equally between wait and propagation time.

Optimistic Adaptation using Gossiped Information: To study the suitability of using gossip-based monitoring to drive adaptation, we examined using gossiping to drive load-balancing decisions in a distributed application. In this system, monitoring is done by having each node track of



(a) NAS Benchmark Effectiveness

(b) Time interval analysis

Figure 8. Effectiveness and Time Interval Metric Measurements for NAS Benchmarks and ChemCell Synthetic Application

its own workload and by gossiping the system-wide maximum and minimum of these local workloads. By comparing the gossiped maximum workload with the local workload using the ratio *maximum / local*, a process can determine if another process is more highly loaded than the local process. Similarly, by comparing the gossiped minimum workload with the local workload using the ratio *local / minimum*, a process can determine whether this process is more highly loaded than other processes. If either of these ratios is high, the process can decide that load imbalance exists, and a load balancing action is needed.

Because of the decentralized nature of Embedded Gossip, however, the workload estimates gathered in this system may by inconsistent. For example, one process may have noticed imbalance and decided that a load balancing action is needed, while others may have seen no imbalance at all. If not dealt with properly, localized load checks may return different results at different processes, which can cause deadlock where some processes wait indefinitely for global synchronous load balancing action, and others continue with their computation.

We address this problem by developing a new simplified variant of the two-phase commit algorithm [4] termed *3-wait* and shown in figure 9. The *3-wait* algorithm, like the standard 2PC protocol, uses a coordinator process and ensures that either all processes commit to perform an action or none do. Unlike 2PC, however, 3-wait assumes that there are that vote-requests happen implicitly through the gossiping of performance information. This prevents applications that do not need to perform any commits (for example, load balancing applications running a data set that never goes out of balance) from paying unnecessary vote request and commit costs.

To investigate the performance of this approach across a range of scenarios, we evaluated conducted additional experiments using the *imb-please* benchmark over a wide range of synthetic load balancing parameters. In particular, we compared speedups between the gossip-based and traditional scheduling of load balancing actions. As shown in Figure 10, gossip-based scheduling of load balancing can effectively detect and initiate load balancing actions, resulting substantial speedups, but the gossip-based system is not as effective as conventional load balancing at this scale. When the global communication is frequent (comm scale 1), the gossip-based and conventional approaches are comparable, but the speedup in the gossip-based approach degrades as the frequency of global communications decreases. This degradation is due to increased inconsistency in measured load imbalance between processes, resulting aborts in the 3-wait algorithm and missed load balancing

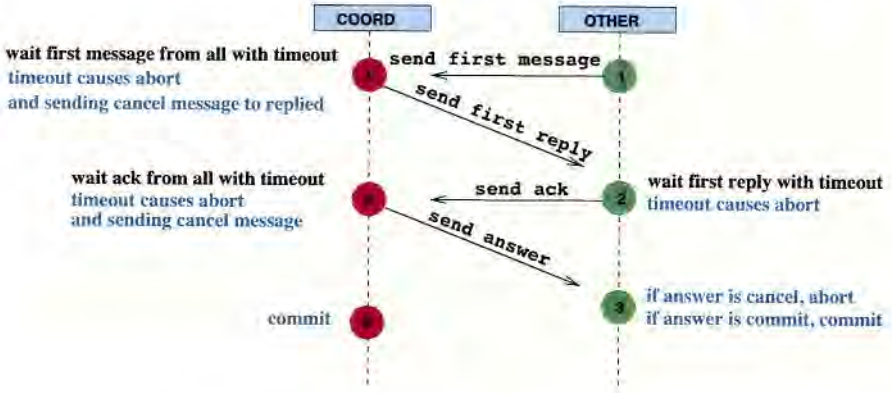


Figure 9. Time flow diagram for the *3-wait* algorithm

opportunities.

Testbeds for Conducting Basic Sensor Network Research: Conducting leading-edge network protocol and data transformation research on sensor networks requires an adequate system for evaluating results; unfortunately, existing simulation and hardware systems are frequently inadequate for this purpose. To support our other research on this project, we evaluated a number of different sensor network platforms, including simulation systems such as TOSSIM and GloMoSim, and have developed a hardware testbed system to use to conduct additional network protocol research on sensor networks. This system provides basic functionality for allocating and programming sensors, and we have recently enhanced this system with new system software extensions that let us control failures in both simulation-based and hardware-based sensor networks, enhancing and enabling future research on network protocols for addressing correlated failures in sensor networks.

References

[1] G. Eisenhauer, "The evpath library." [Online]. Available: `http://www.cc.gatech.edu/systems/projects/EVPath`

[2] M. M. Hayat. J. E. Pezoa, D. Dietz, and S. Dhakal, "Dynamic load balancing for robust distributed computing in the presence of topological impairments," *Wiley Handbook of Science and Technology for Homeland Security*, 2009.

[3] R. Horn and C. Johnson, *Matrix analysis.* Cambridge University Press 1999.

[4] B. W. Lampson and H. Sturgis. Crash recovery in a distributed data storage system. Technical report, Computer Science Laboratory, Xerox, Palo Alto Research Center, Palo Alto, CA, 1976.

[5] Y. Mostofi, "Binary Consensus with Gaussian Communication Noise: A Probabilistic Approach," Proceedings of the 46th IEEE Conference on Decision and Control (CDC), Dec. 2007

[6] Y. Mostofi and Y. Ruan, "Binary Consensus over AWGN Channels," in revision, IEEE Transactions on Automatic Control, Dec. 2008.

[7] J. E. Pezoa, S. Dhakal and M. M. Hayat, "Decentralized Load Balancing for Improving Reliability in Heterogeneous Distributed Systems." In *Proc. of The International Workshop on Design, Optimization and Management of Heterogeneous Networked Systems (DOM-HetNetS '09), Vienna, Austria, September 22-25, 2009.*

[8] J. E. Pezoa, S. Dhakal and M. M. Hayat, "Maximizing service reliability in distributed computing systems with random failures: Theory and implementation," *under minor review in IEEE Trans. Parallel and Dist. Systems*, 2009.

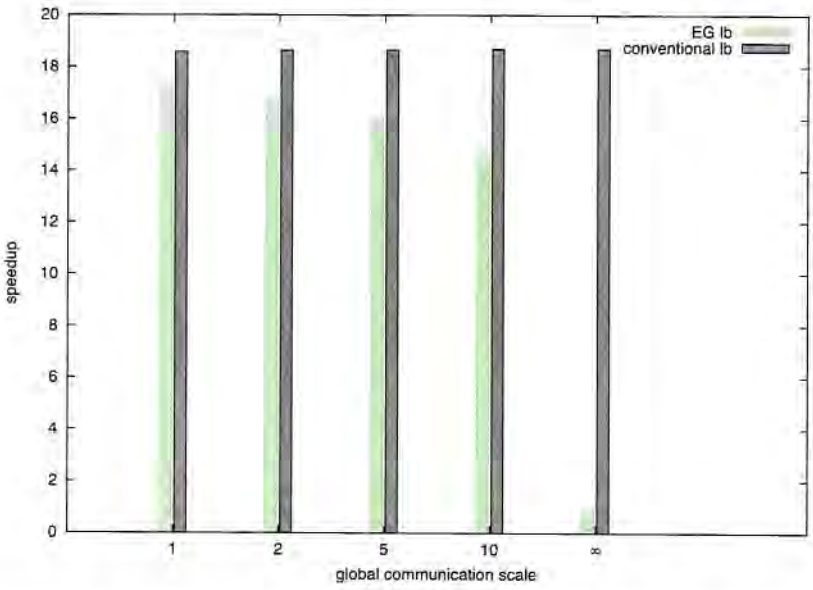


Figure 10. Comparison of different global communication scales, using *imb-please*, with single overloading, overload factor 3.

[9] W. Ren and R. W. Beard and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," Proceedings of 24th American Control Conference, 2005.

[10] W. Ren and R. W. Beard and E. M. Atkins, "Information consensus in multivehicle cooperative control," IEEE Control Systems Magazine, April 2007.

[11] Y. Ruan and Y. Mostofi, "Binary Consensus with Soft Information Processing in Cooperative Networks," Proceedings of the 47th IEEE conference on decision and control, 2008.

[12] Y. Yuan and Y. Mostofi, "Impact of Link Qualities and Network Topology on Binary Consensus," Proceedings of the 2009 American Control conference, 2009.

[13] Wenbin Zhu, Patrick G. Bridges, and Arthur B. Maccabe. Online critical path profiling for parallel applications. In *Proceedings of the 2005 IEEE International Conference on Cluster Computing (Cluster 2005)*, Boston, MA, September 2005.

[14] Wenbin Zhu, Patrick G. Bridges, and Arthur B. Maccabe. Embedded gossiping: Lightweight online measurement for large-scale applications. In *Proceedings of the 2007 IEEE International Conference on Distributed Computing Systems (ICDCS)*, June 2007.

[15] Wenbin Zhu, Patrick G. Bridges, and Arthur B. Maccabe. Lightweight application monitoring and tuning with embedded gossip. *IEEE Transactions of Parallel and Distributed Systems (TPDS)*, 20(7):1038–1049, July 2009.

Appendix A. Algorithm

Algorithm 1 Algorithm to devise task reallocation policies for multi-server DCSs

Require: κ, $\hat{m}_{j,i}$ and $K_{ij}^{(0)}$, with $j = 1, \ldots, n$, $i \neq j$
Ensure: K_{ij}
  Set $U_i = \{j : K_{ij}^{(0)} > 0\}$, $U_i^{'} = \emptyset$ and $k = 1$
  loop
    while $j \in U_i$ **do**
      $U_i \leftarrow U_i \setminus \{j\}$
      $m_1 = m_i - \sum_{\ell \in U_i} L_{i\ell}^{(k-1)} - \sum_{\ell \in U_i^{'}} L_{i\ell}^{(k)}$ and $m_2 = \hat{m}_{j,i}$
      Solve (3) using m_1 and m_2 to obtain $K_{ij}^{(k)}$
      $U_i^{'} \leftarrow U_i^{'} \cup \{j\}$
    end while
    Set $U_i = \{j : K_{ij}^{(0)} > 0\}$, $U_i^{'} = \emptyset$ and $k \leftarrow k + 1$
    if $\sum_{j=1}^{n} \left(K_{ij}^{(k)} - K_{ij}^{(k-1)}\right) = 0$ or $k > \kappa$ **then**
      $K_{ij} = K_{ij}^{(k)}$ for all $j \in U_i$ and **exit**
    end if
  end loop

DEPARTMENT OF DEFENSE

DEFENSE TECHNICAL
INFORMATION CENTER
8725 JOHN J. KINGMAN ROAD,
SUITE 0944
FT. BELVOIR, VA 22060-6201
ATTN: DTIC/OCA

DEPARTMENT OF DEFENSE CONTRACTORS

ITT INDUSTRIES
ITT SYSTEMS CORPORATION
1680 TEXAS STREET, SE
KIRTLAND AFB, NM 87117-5669
ATTN: DTRIAC